A code for the study of granular and rubble-pile dynamics with non-spherical particles

Dr. Fabio Ferrari

Jet Propulsion Laboratory, California Institute of Technology

394A – Mission Engineering & Planning

Presentation overview

1. Introduction
   - Research goal and motivation

2. Implementation and methods
   - Software architecture
   - Gravitational dynamics
   - Contact dynamics
   - Numerical integration

3. Applications
   - Rubble-pile asteroid
   - Granular soil interaction

4. Conclusion
   - Final highlights
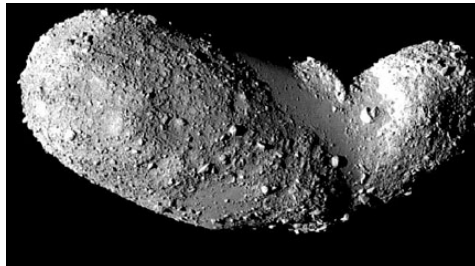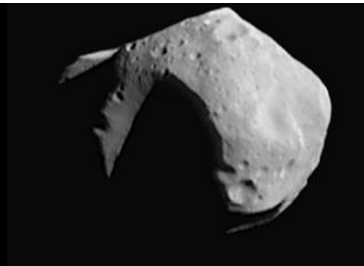   - Future work and ongoing collaborations

JPL

Introduction

Motivation and research goal

Credits: JAXA/Hayabusa   Credits: NASA/NEAR

Many asteroids between 100 m and 100 km in size are likely to be gravitational aggregates "rubble pile" [Richardson et al. 2002]

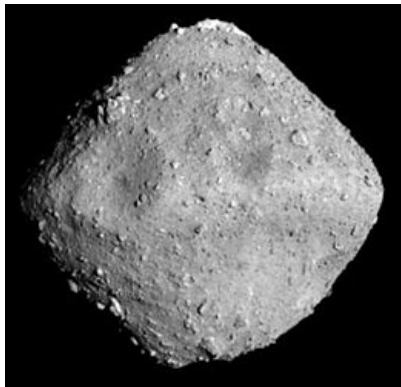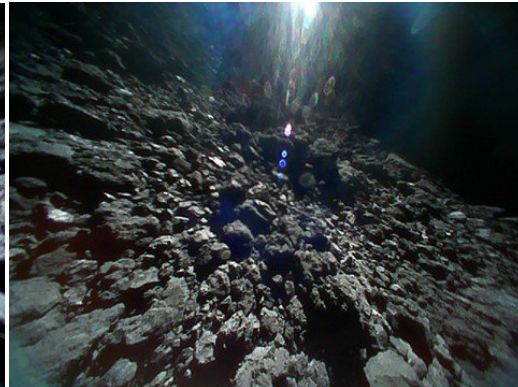**Research goal**:

Study of rubble-pile asteroids as gravitational aggregates through numerical simulations (granular dynamics)
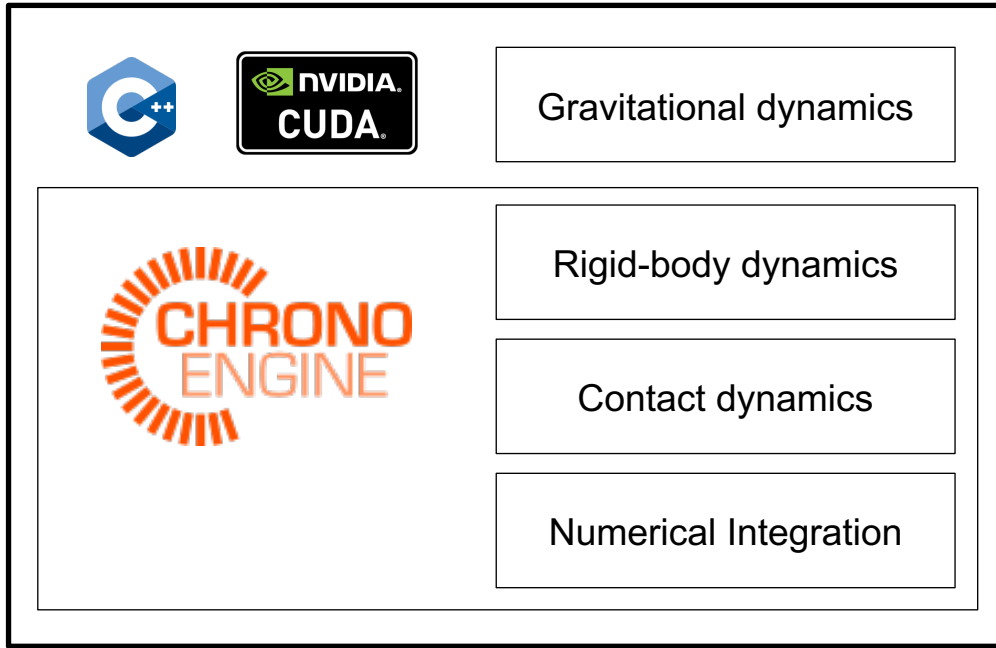

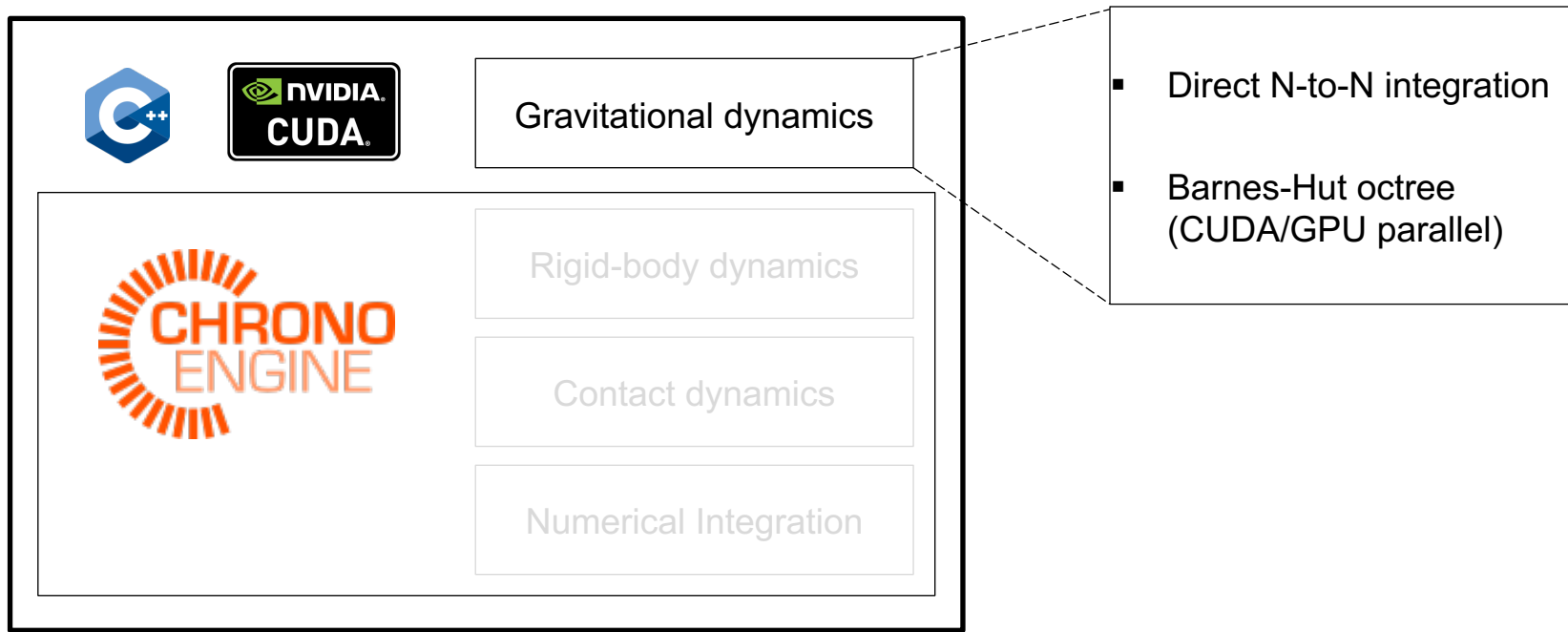
Credits: JAXA/
Hayabusa 2/
Minerva-II1-B

JPL

Implementation and methods

Implementation and methods
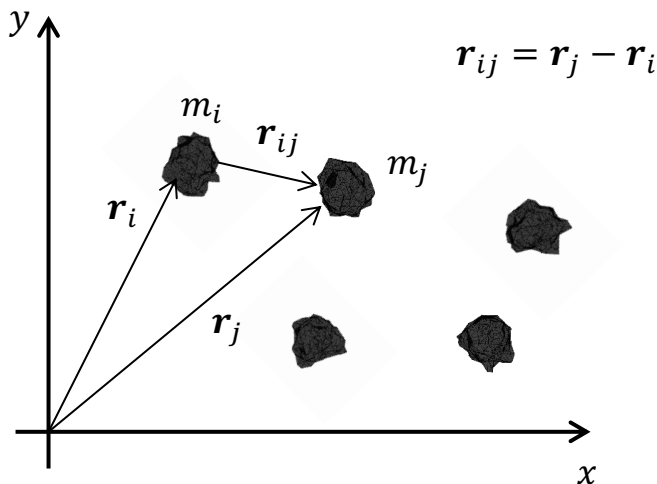
Software architecture

Implementation and methods

Gravitational dynamics

Gravitational dynamics

Rigid-body dynamics

Contact dynamics

Numerical Integration

- Direct N-to-N integration

- Barnes-Hut octree (CUDA/GPU parallel)

Implementation and methods

Gravitational dynamics: direct N-to-N integration

$$\boldsymbol{r}_{ij} = \boldsymbol{r}_j - \boldsymbol{r}_i$$

N equations of motion

$$m_i \, \ddot{\boldsymbol{r}}_i = G \sum_{j=1, j \neq i}^{N} \frac{m_i \, m_j}{r_{ij}^3} \boldsymbol{r}_{ij}$$
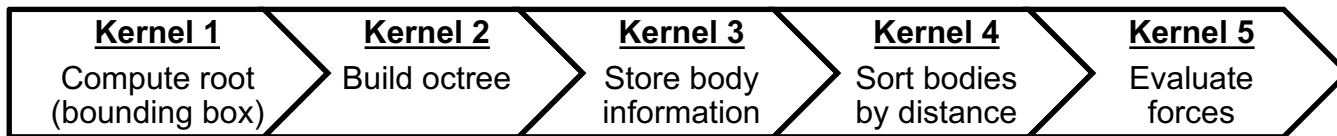
Features of the dynamical system

- No analytical solution for the gravitational motion of N bodies
- Highly non-linear (chaotic) behavior
- Strong dependency on initial conditions
- Slow dynamics: characteristic time $T \sim \frac{1}{\sqrt{G\rho}}$

  (with $G = 6.67 \cdot 10^{-11} \frac{m^3}{kg \, s^2}$)

Features of the numerical problem

- Initial value problem
- Integration time step can be big
  $dt < \frac{T}{2} = \frac{1}{2\sqrt{G\rho}}$
  ($dt \sim 10^3 \, s$ for typical asteroids densities)

JPL

Implementation and methods

Gravitational dynamics: Barnes-Hut octree (CUDA/GPU parallel)

Kernel 1 Compute root (bounding box)	**Kernel 2** Build octree	**Kernel 3** Store body information	**Kernel 4** Sort bodies by distance	**Kernel 5** Evaluate forces

Based on the work by M. Burtscher and K. Pingali, *GPU Computing Gems*, Chapter 6: An efficient CUDA implementation of the Tree-based Barnes Hut N-body Algorithm. Elsevier Inc, 2011.

Root
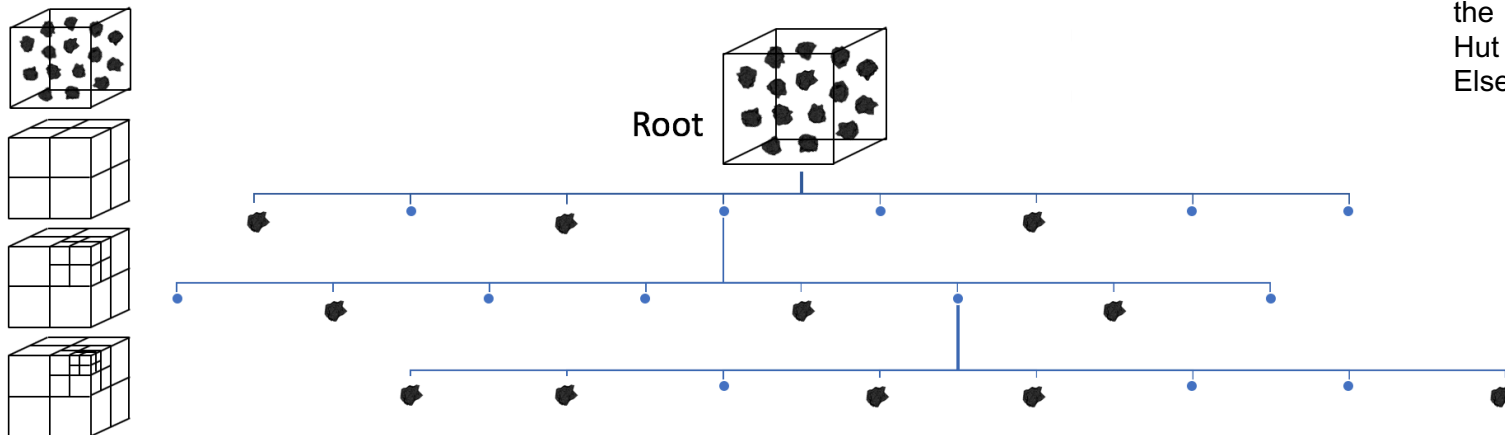
- **Nodes** correspond to **cubes** in the physical space
- Homogenous Spatial Recursive sub-division (until each extremal node has 1 or 0 particles)

 JPL

Implementation and methods

Gravitational dynamics: Barnes-Hut octree (CUDA/GPU parallel)

Under certain conditions, the force acting on a body, generated by a cluster of bodies can be approximated ➔ treat cluster as a single body

For each Body-Node pair $(\boldsymbol{B}, \boldsymbol{N})$:
$$\theta = \frac{Radius\ of\ N}{\boldsymbol{R_N} - \boldsymbol{R_B}}$$

$\boldsymbol{R_N}$: position of barycenter of node $\boldsymbol{N}$ of the octree.

$\boldsymbol{R_B}$: position of body $\boldsymbol{B}$.

After choosing the accuracy ($\theta_{accuracy}$) the condition is:
$$\theta < \theta_{accuracy}$$

- $\theta_{accuracy} = 0$ is the limiting case of considering all interactions between bodies
- Typical value: $\theta_{accuracy} = 0.25$ (the body-to-cluster distance is at least 4 times the radius of the cluster)

JPL

Implementation and methods

Gravitational dynamics: Barnes-Hut octree (CUDA/GPU parallel)

For each body **B**, the tree is traversed from the root downwards.

<u>Every time a node **N** is encountered:</u>

- **If *N* is a leaf:**
  body-to-body interaction

- **If *N* is internal and $\theta < \theta_{accuracy}$:**
  Traversal interrupted and body-to-cluster interaction

- **If *N* is internal and $\theta \geq \theta_{accuracy}$:**
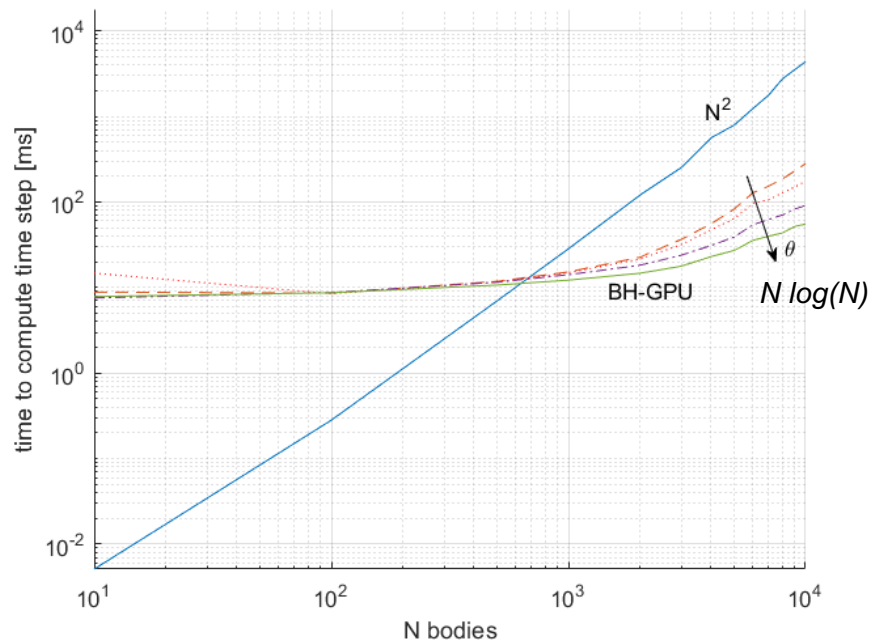 Traversal continues

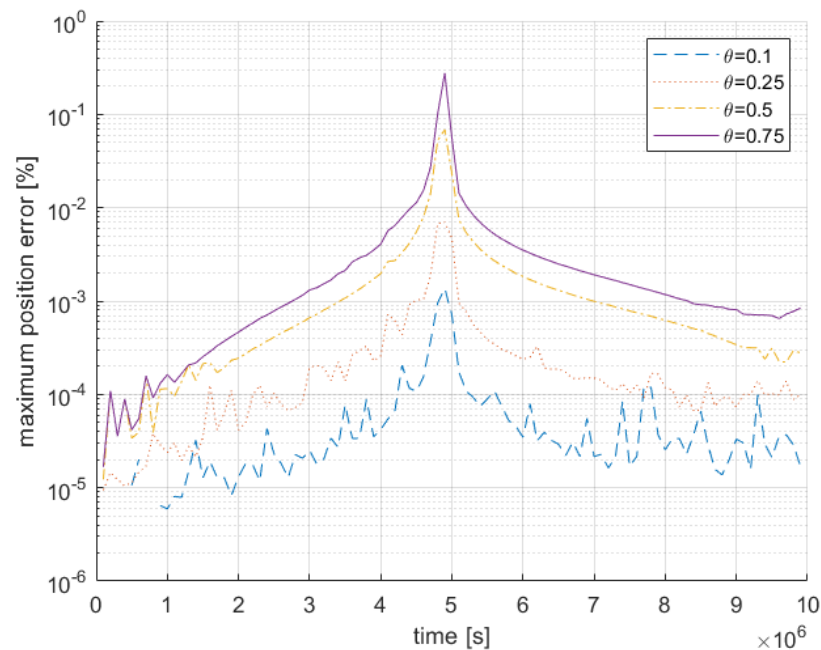# Implementation and methods

Gravitational dynamics: performance

CPU: Intel Core i7 6500U 3.1GHz
GPU: Nvidia GeForce 940M

Computational time
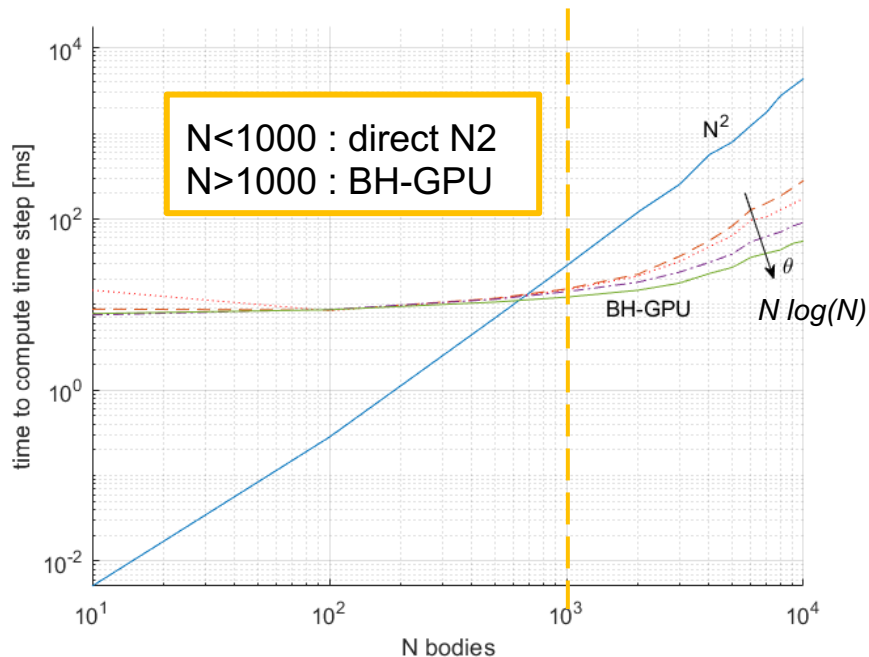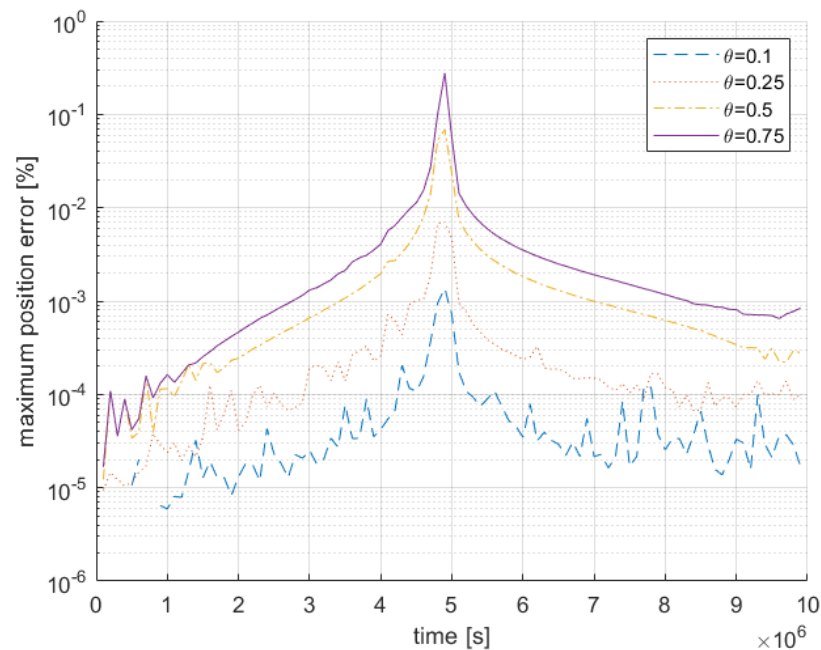
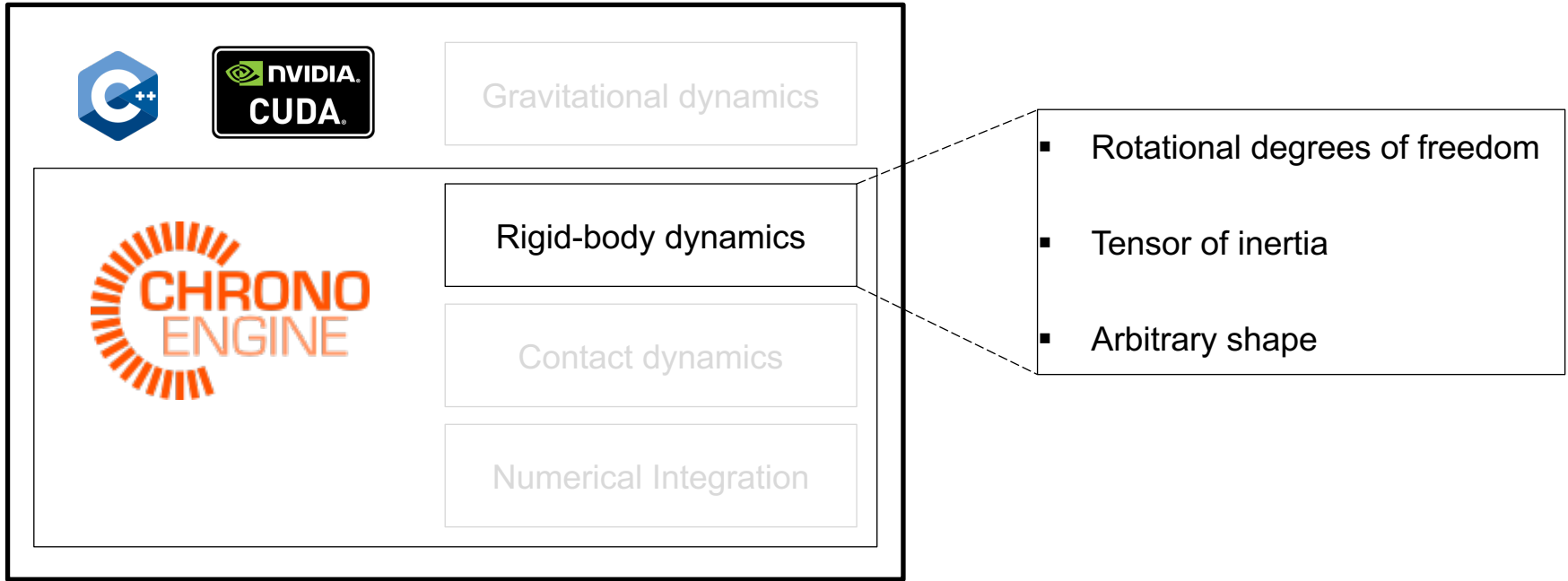Accuracy (depends on $\theta_{accuracy}$)

JPL

Implementation and methods

Gravitational dynamics: performance

Computational time

Accuracy (depends on $\theta_{accuracy}$)

N<1000 : direct N2
N>1000 : BH-GPU

N^2

BH-GPU

$N \log(N)$

θ

time to compute time step [ms]

N bodies

maximum position error [%]

time [s]

$\theta=0.1$
$\theta=0.25$
$\theta=0.5$
$\theta=0.75$

JPL

Implementation and methods

Rigid-body dynamics

- Rotational degrees of freedom
- Tensor of inertia
- Arbitrary shape

Boxes: Gravitational dynamics, Rigid-body dynamics, Contact dynamics, Numerical Integration

Implementation and methods

Rigid-body dynamics

N bodies, each with

- position $\boldsymbol{r}_i$
- rotation quaternion $\boldsymbol{\rho}_i$
- velocity $\dot{\boldsymbol{r}}_i$
- angular velocity $\boldsymbol{\omega}_i$

Generalized coordinates
$$\boldsymbol{q} = \left\{ \boldsymbol{r}_i^T, \boldsymbol{\rho}_i^T \right\}^T \in \mathbb{R}^{7N}$$
$$\boldsymbol{v} = \left\{ \dot{\boldsymbol{r}}^T{}_i, \boldsymbol{\omega}_i^T \right\}^T \in \mathbb{R}^{6N}$$
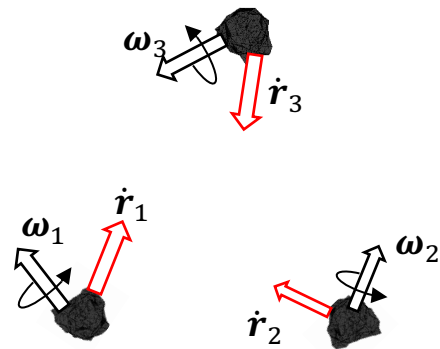
- mass m_i
- tensor of inertia $\mathbf{I}_i$

$$\mathbf{M} = [m_i] \in \mathbb{R}^{6N \times 6N}$$
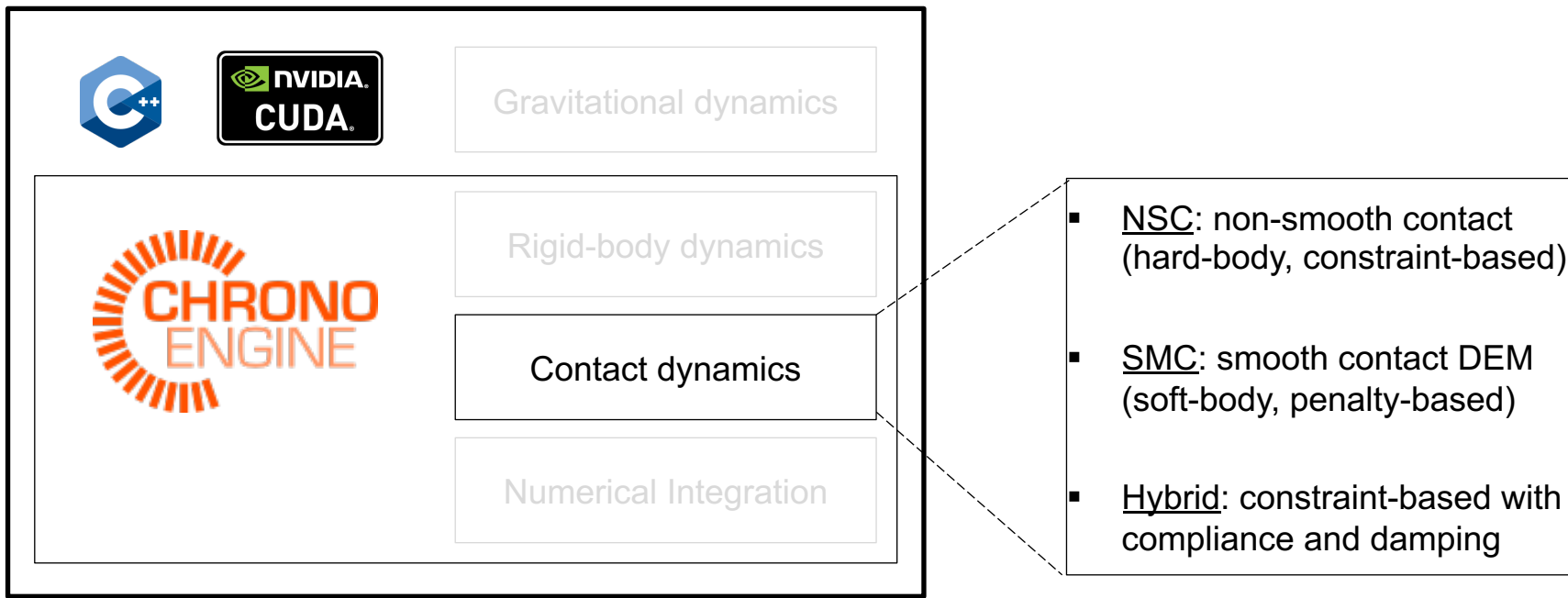$$\mathbf{J} = [\mathbf{I}_i] \in \mathbb{R}^{6N \times 6N}$$

- collision surface Ω_i

Shape:
- Triangulated mesh
- Convex hull
- Common geometry (sphere, box, cone,…)

JPL

Implementation and methods

Contact dynamics

- **NSC**: non-smooth contact (hard-body, constraint-based)

- **SMC**: smooth contact DEM (soft-body, penalty-based)

- **Hybrid**: constraint-based with compliance and damping

JPL

Implementation and methods

Contact dynamics: non-smooth dynamics (NSC)

- Equations of motion are formulated as Differential Variational Inequalities (DVI)
- Hard-body model
- Complementarity-based
- Impulse-momentum formulation
- Suitable for problems with discontinuities (rigid contacts)

$$\begin{cases} \gamma \text{ (contact) as solution of CCP} \\ \boldsymbol{v_{n+1}} = f(\boldsymbol{q}, \boldsymbol{v}, t, \gamma) \\ \boldsymbol{q_{n+1}} = g(\boldsymbol{q}, \boldsymbol{v}) \end{cases}$$

Parameters of the model:
- Friction (static, dynamic, spinning)
- Cohesion (value and constitutive model)
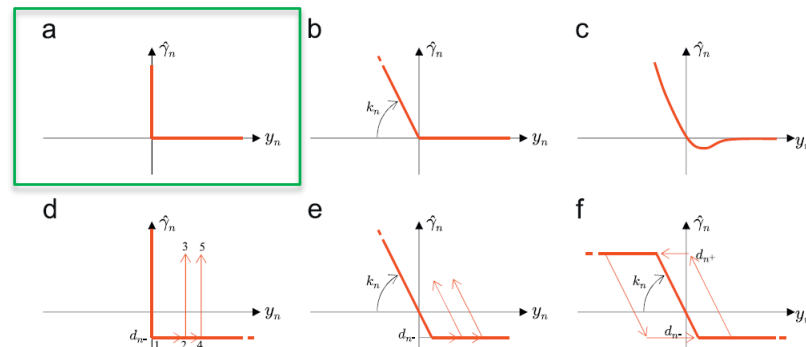- Restitution coefficient

Credits: Tasora et al 2013

Fig. 1. Basic constitutive relations for normal reaction.

Implementation and methods

Contact dynamics: smooth dynamics (SMC)

- Equations of motion are formulated as Differential Algebraic equations (DAE)

$$\begin{cases} \dot{\boldsymbol{x}} = f(\boldsymbol{x}, t) \\ \boldsymbol{g}(\boldsymbol{x}, t) = 0 \end{cases}$$

ODE + AE (kinematic constraint)

- Soft-body model (DEM)
- Penalty-based
- Force-acceleration formulation
- Suitable for problems with no discontinuities (no rigid contacts)

Credits: Tasora et al 2013

Parameters of the model:

- Friction (static, dynamic, spinning)
- Cohesion (value and constitutive model)
- {Young modulus, Poisson ratio, restitution coefficient} or {stiffness and damping (normal and tangential)} and constitutive model (Hooke, Hertz)

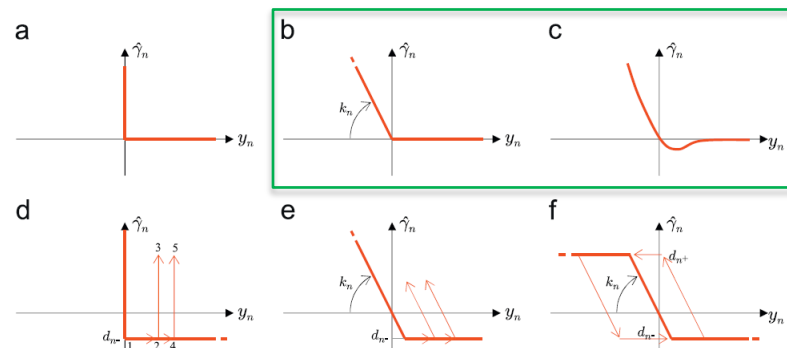In this case stiffness and damping are estimated based on constitutive law of material

Fig. 1. Basic constitutive relations for normal reaction.

Implementation and methods

Contact dynamics: hybrid model

$$\begin{cases} \gamma \ (\text{contact}) \ \text{as solution of CCP} \\ \boldsymbol{v_{n+1}} = f(\boldsymbol{q}, \boldsymbol{v}, t, \gamma) \\ \boldsymbol{q_{n+1}} = g(\boldsymbol{q}, \boldsymbol{v}) \end{cases}$$

- Equations of motion are formulated as Differential Variational Inequalities (DVI)
- Soft-body model (compliance and damping)
- Complementarity-based
- Impulse-momentum formulation
- Suitable for problems with discontinuities

Credits: Tasora et al 2013

Parameters of the model:

- Friction (static, dynamic, spinning)
- Cohesion (value and constitutive model)
- Restitution coefficient
- Stiffness and damping (normal, tangential, rolling, spinning), rolling friction and constitutive model

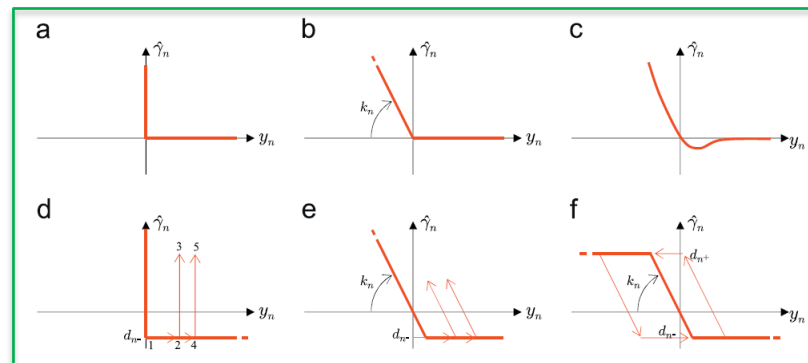**Fig. 1.** Basic constitutive relations for normal reaction.

JPL

Implementation and methods

Contact dynamics: summary

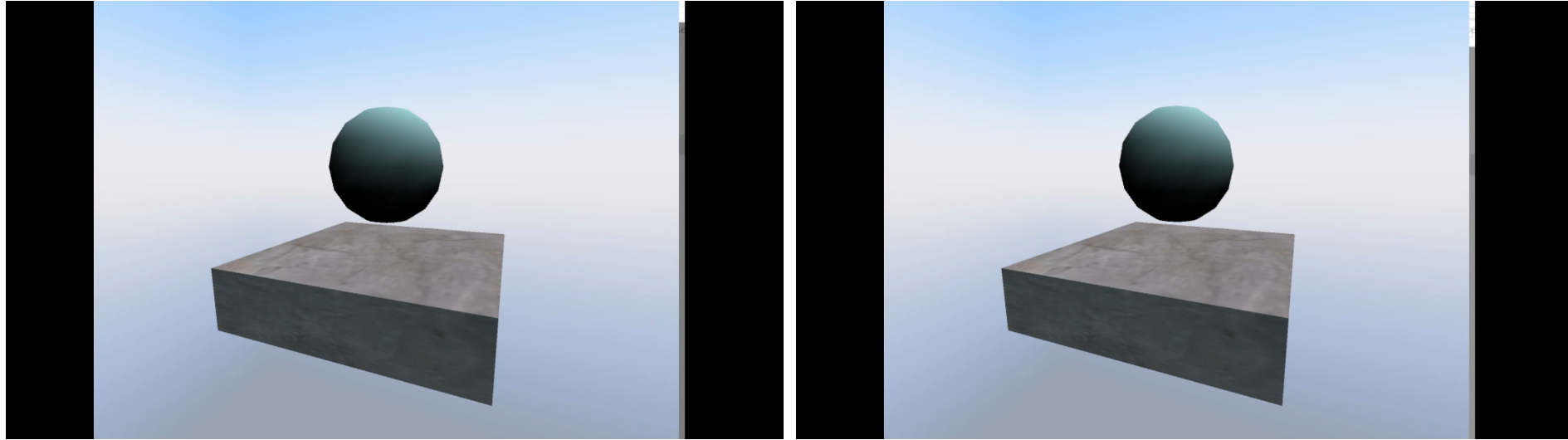|		NSC	SMC	Hybrid
Formulation	Equations of motion	DVI	DAE	DVI
	Contact model	hard	soft	soft
Performance	Computational time (single time step)	🟥	🟩	🟥
	Size of time step	🟩	🟥	🟨
	Reproducing non-rigid contact dynamics	🟥	🟩	🟩
	Handling complex shapes	🟩	🟥	🟩

JPL

Implementation and methods

Contact dynamics: tuning the parameters

Hybrid model

SMC (DEM)
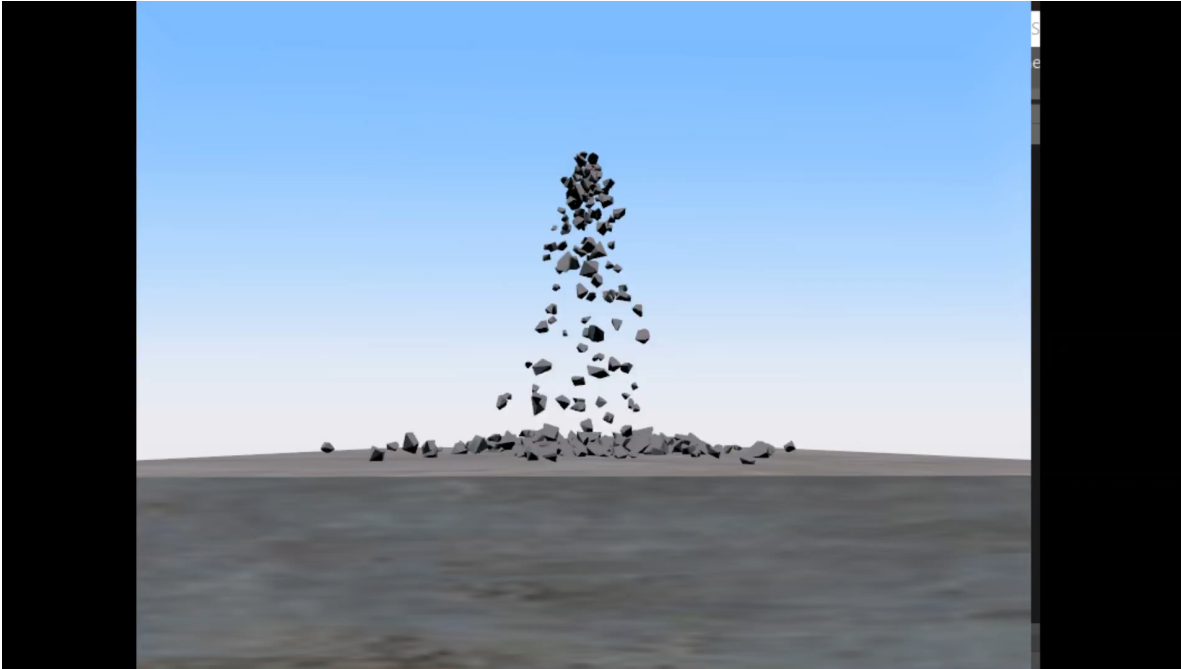
JPL

Implementation and methods

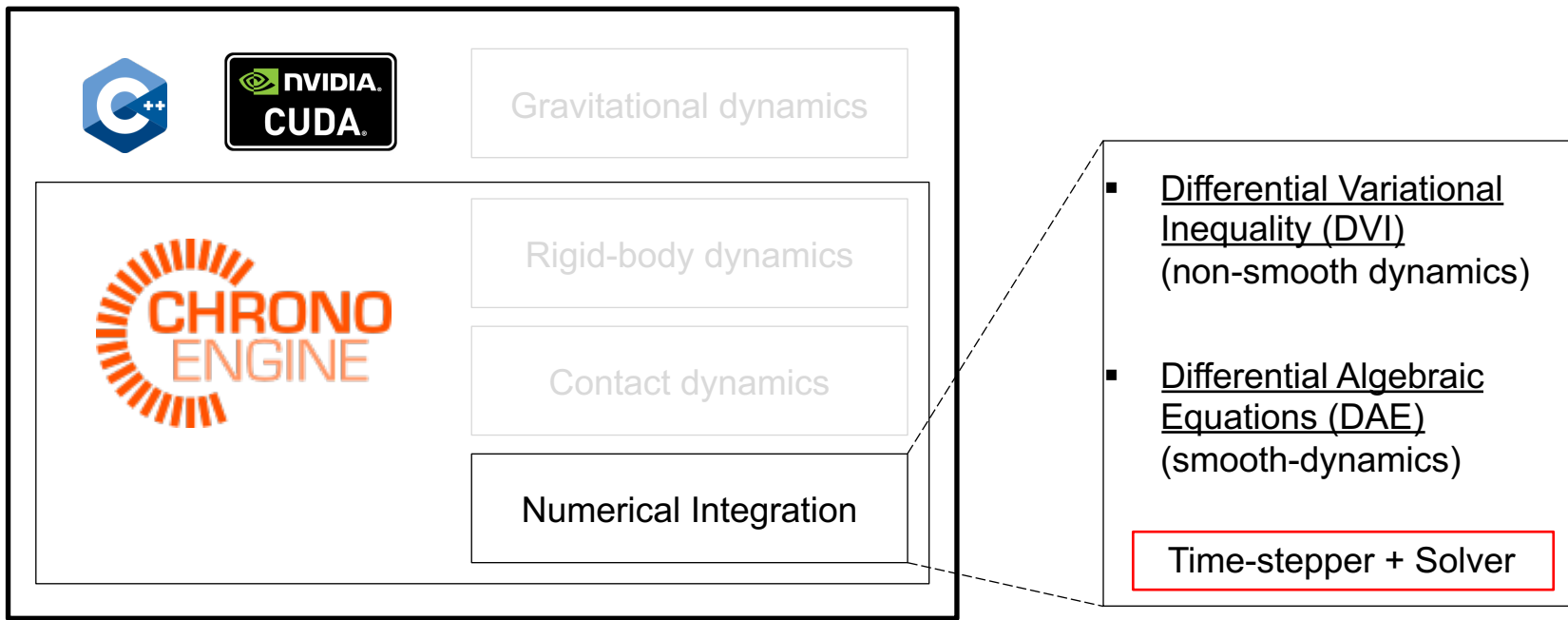Contact dynamics: tuning the parameters

Angle of repose

JPL

Implementation and methods

Numerical integration

Gravitational dynamics

Rigid-body dynamics

Contact dynamics

Numerical Integration

- Differential Variational Inequality (DVI) (non-smooth dynamics)

- Differential Algebraic Equations (DAE) (smooth-dynamics)

Time-stepper + Solver

Implementation and methods

Numerical integration: available methods

Non-smooth dynamics (NSC)

Equations of motion are formulated as <u>Differential Variational Inequalities (DVI)</u>

Smooth dynamics (SMC)

Equations of motion are formulated as a <u>Differential Algebraic Equations (DAE)</u>

- Time-steppers:
  - Symplectic methods (semi-implicit Euler, leapfrog)   → Suited for gravitational problem
  - Runge Kutta methods (RK45, explicit Euler, implicit Euler, trapezoidal, Heun) → Higher order
  - Newmark, Hilber-Hughes-Taylor   → Suited for FEA problems

- Solvers:
  - Iterative solvers
  - Direct solvers

Most commonly used: good for both DVI and DAE problems

Applications

Applications

Rubble-pile asteroid: aggregation

JPL

Applications

Rubble-pile asteroid: aggregation

Parameter	Symbol
Linear relative velocity of bodies	v_0
Angular relative velocity of bodies	ω_0
Orbital angular momentum of bodies	L_0
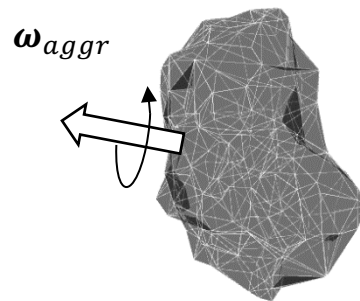
$\boldsymbol{\omega}_{aggr}$

Aggregation dynamics

- State of bodies
- Aggregation time
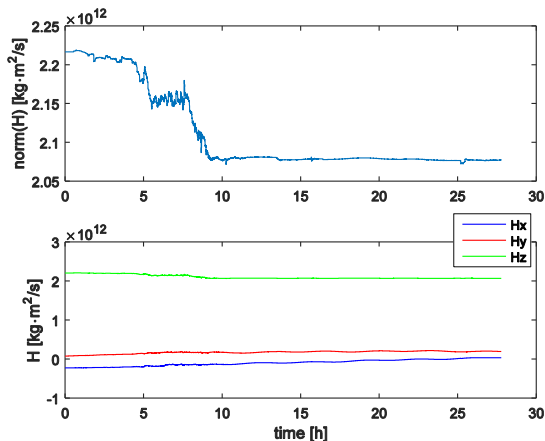- Angular momenta profile

Physical properties (final aggregate)

- Bulk density / void fraction
- Shape: inertia elongation
- N bodies in aggregate
- Total mass and size

Dynamical state (final aggregate)

- Angular velocity
- Angular momentum
- Breakup limits

Applications

Rubble-pile asteroid: aggregation

Orbital energy is transmitted to spinning rotations of single bodies through to collisions

$$v_0 = \omega_0 = 0; \quad L_0 \neq 0$$
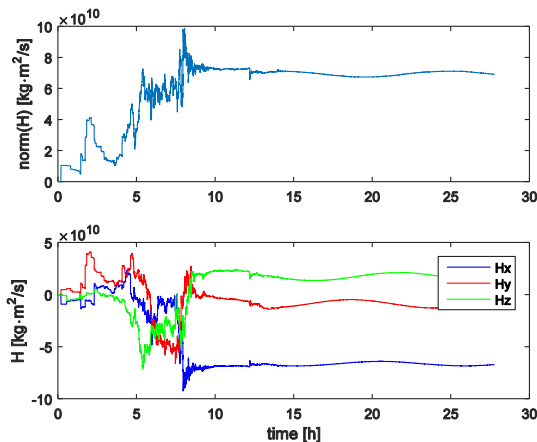
Spinning angular momenta of single bodies is transmitted to to the aggregate through collisions

$$v_0 = L_0 = 0; \quad \omega_0 \neq 0$$
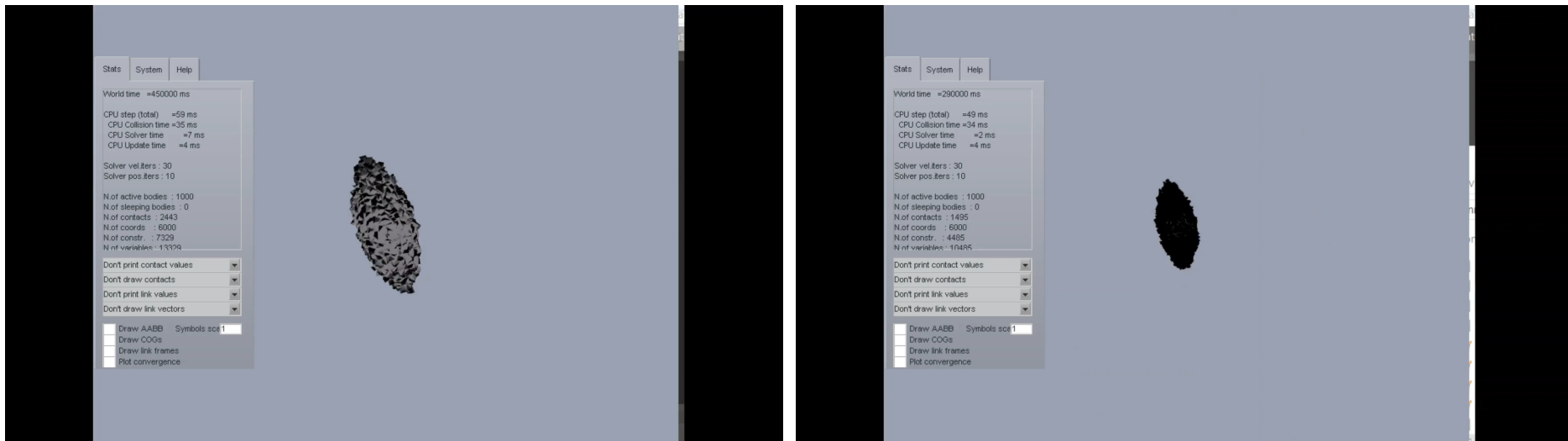
Shape/elongation

$$\lambda = \frac{I_{max}}{I_{min}} = [1.15 - 2.70]$$

Porosity
- Medium-size aggregates: $[34\% - 40\%]$
  $(\rho_b \cong 1900 \frac{kg}{m^3})$
- Small aggregates: $[14\% - 18\%]$
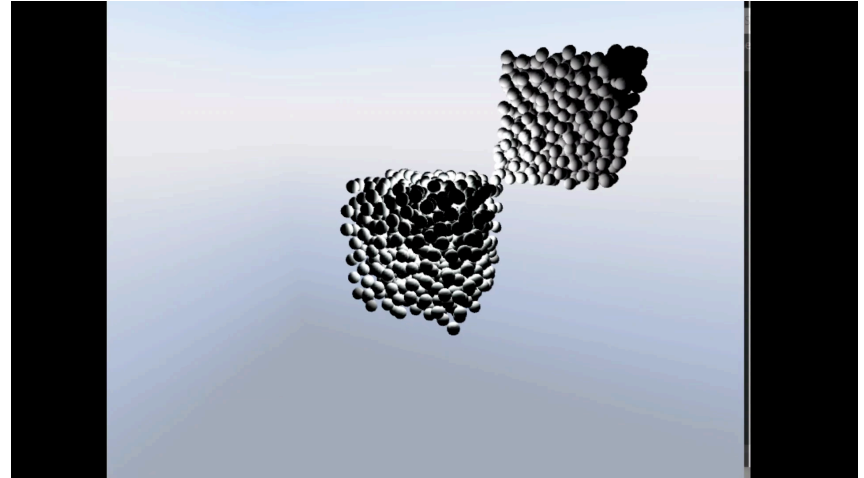 $(\rho_b \cong 2500 \frac{kg}{m^3})$

Applications

Rubble-pile asteroid: spin-up

JPL

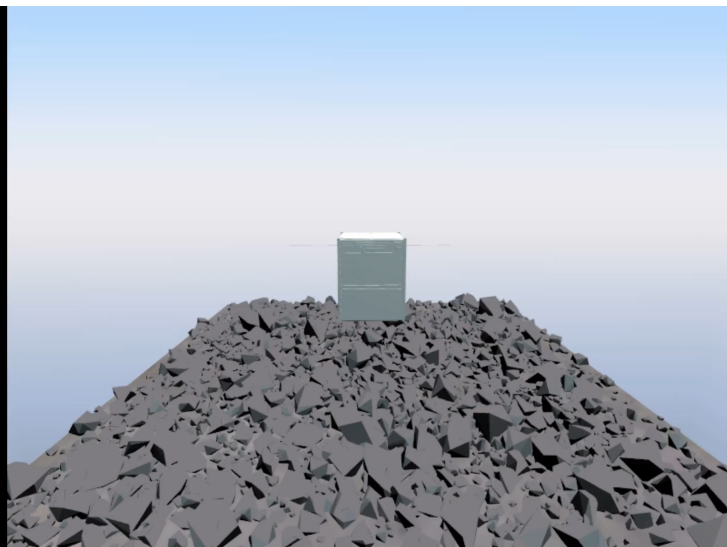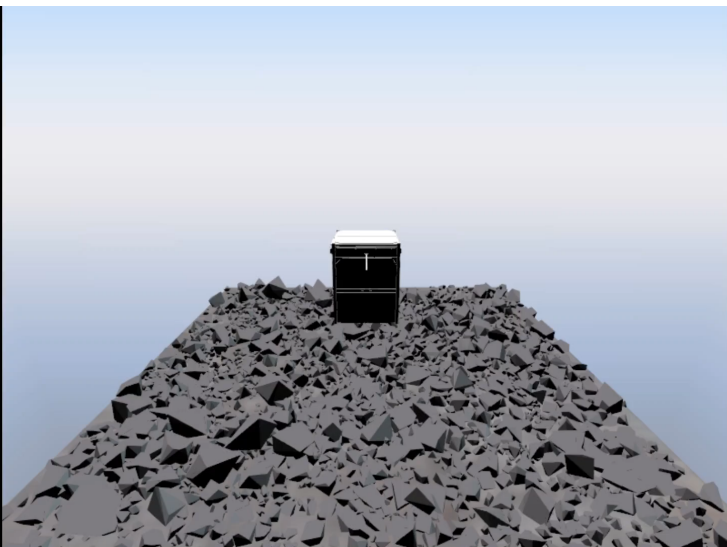# Applications

More scenarios

- Tidal disruption of rubble-pile
- Collision between rubble-piles
- High-velocity impacts
  (co-simulation with SPH)
- Rubble-pile model as high-fidelity
  gravity source model
- Planetary ring dynamics
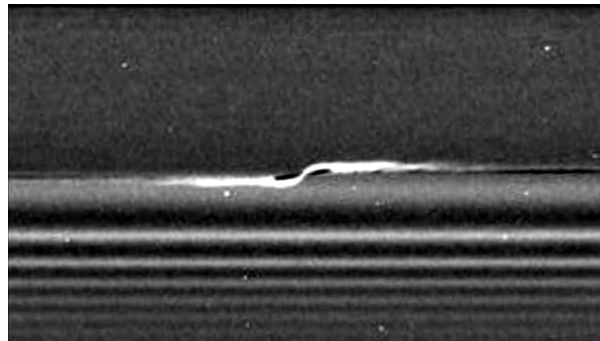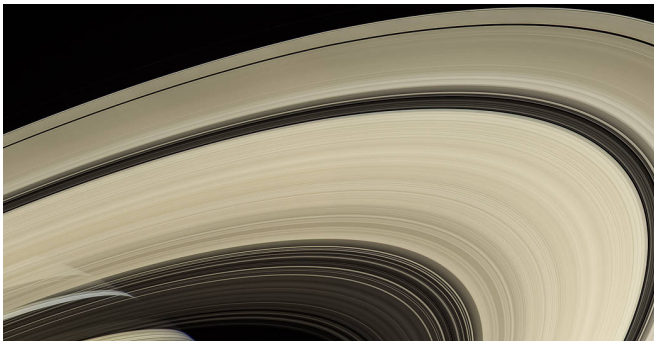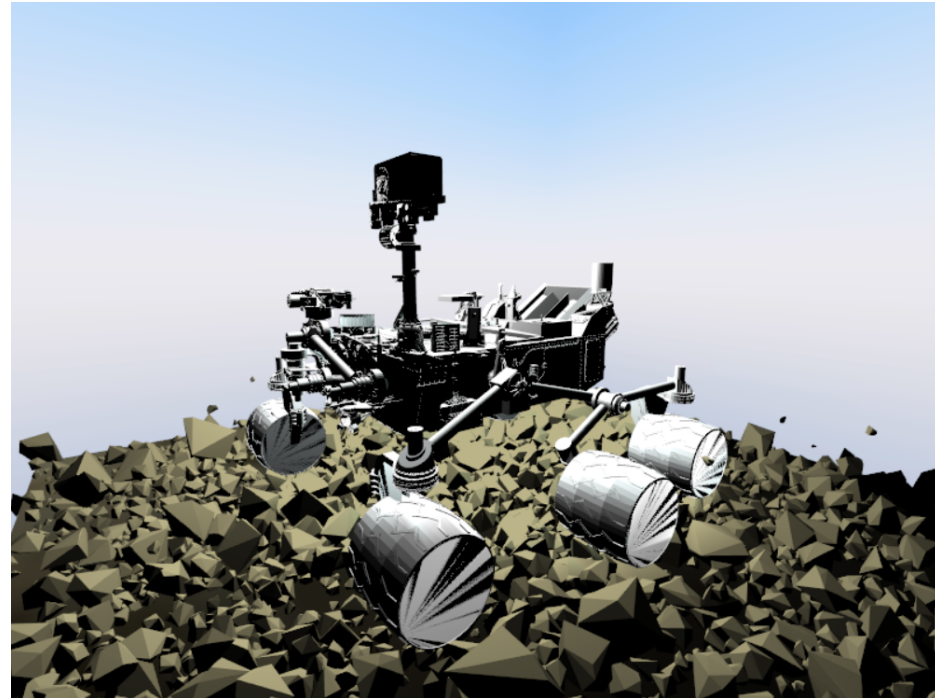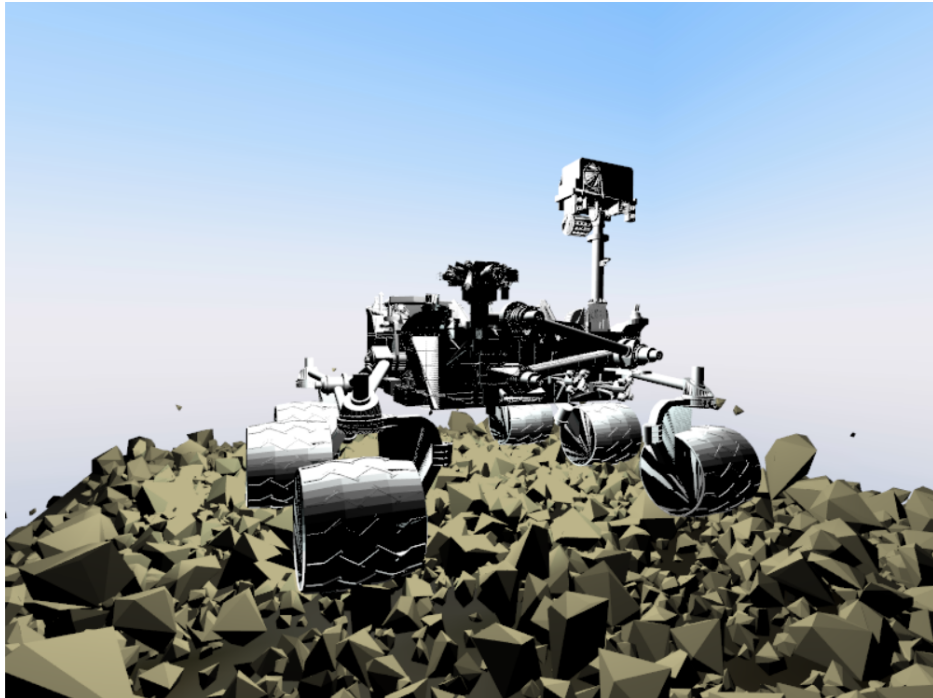

Credits: NASA/JPL/Cassini

JPL

Applications

Granular soil interaction

© 2018 California Institute of Technology. Government sponsorship acknowledged.

Applications

Granular soil interaction

JPL

Conclusion

FINAL HIGHLIGHTS

- Handles complex-shaped bodies
- State-of-the-art methods for gravitational dynamics: Barnes-Hut parallel GPU
- State-of-the-art methods for contact dynamics: both hard- and soft-contact models
- Great flexibility of models/methods and implementation

FUTURE WORK AND ONGOING COLLABORATIONS

- Go on with validation/benchmarking and developing effort (with Chrono::Engine team, Univ. Parma)
- Rubble pile aggregation / reconfiguration (with OCA)
- Lander/soil interaction and lander/rover mobility
- Planetary rings dynamics
- Rubble pile gravity field

JPL

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov